Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	9	compromis$5 and "leaf node" and "internal node" and key$1 and leaf adj2 node$1 and encrypt$4 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:35
L2	43	compromis$5 and ("leaf node" or leaf adj2 node$1) and key$1 and encrypt$4 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:29
L3	10	compromis$5 and ("leaf node" or leaf adj2 node$1) and key$1 and encrypt$4 and internal adj2 node$1 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 11:34
L4	13	compromis$5 same (leaf or node$) and ("leaf node" or leaf adj2 node$1) and internal adj2 node$1 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:29
L5	2084	707/200.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 11:35
L6	0	5 and "leaf node" same hierarch$6 same position	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:00
L7	19	5 and "leaf node" and hierarch$6 and position$1	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 11:48
L8	7	7 and "root node"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 11:48
L9	5	5 and "leaf node" with key$1 and child near node$1	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:33
L10	14	leaf near node same key and level adj defin$5	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:04
L11	24	"leaf node" and "internal node" and key$1 and leaf adj2 node$1 and encrypt$4 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:29
L12	168	("leaf node" or leaf adj2 node$1) and key$1 and encrypt$4 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:29

L13	474	(leaf or node$) and ("leaf node" or leaf adj2 node$1) and internal adj2 node$1 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:30
L14	180	13 and "leaf node" and position$1 and path$1	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:33
L15	107	14 and hierarch$6	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:33
L16	13	15 and "leaf node" with key$1 and child near node$1	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:34
L17	36	defin$5 same path$1 same "leaf node" same "root node" and key$1 and leaf adj2 node$1 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:37

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	9	compromis$5 and "leaf node" and "internal node" and key$1 and leaf adj2 node$1 and encrypt$4 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:35
L2	43	compromis$5 and ("leaf node" or leaf adj2 node$1) and key$1 and encrypt$4 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:29
L3	10	compromis$5 and ("leaf node" or leaf adj2 node$1) and key$1 and encrypt$4 and internal adj2 node$1 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 11:34
L4	13	compromis$5 same (leaf or node$) and ("leaf node" or leaf adj2 node$1) and internal adj2 node$1 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:29
L5	2084	707/200.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 11:35
L6	0	5 and "leaf node" same hierarch$6 same position	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:00
L7	19	5 and "leaf node" and hierarch$6 and position$1	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 11:48
L8	7	7 and "root node"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 11:48
L9	5	5 and "leaf node" with key$1 and child near node$1	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:33
L10	14	leaf near node same key and level adj defin$5	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:04
L11	24	"leaf node" and "internal node" and key$1 and leaf adj2 node$1 and encrypt$4 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:29
L12	168	("leaf node" or leaf adj2 node$1) and key$1 and encrypt$4 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:29

L13	474	(leaf or node$) and ("leaf node" or leaf adj2 node$1) and internal adj2 node$1 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:30
L14	180	13 and "leaf node" and position$1 and path$1	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:33
L15	107	14 and hierarch$6	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:33
L16	13	15 and "leaf node" with key$1 and child near node$1	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:34
L17	36	defin$5 same path$1 same "leaf node" same "root node" and key$1 and leaf adj2 node$1 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 12:37
S1	1636	707/200.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 11:35
S2	0	compromis$5 same "leaf node" and "internal node" and key$1 and leaf adj2 node$1 and encrypt$4 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2004/06/23 19:57
S3	9	compromis$5 and "leaf node" and "internal node" and key$1 and leaf adj2 node$1 and encrypt$4 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 11:34
S4	0	compromis$5 same ("leaf node" or leaf adj2 node$1) and key$1 and encrypt$4 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2004/06/24 16:39
S5	39	compromis$5 and ("leaf node" or leaf adj2 node$1) and key$1 and encrypt$4 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 11:34
S6	0	compromis$5 and ("leaf node" or leaf adj2 node$1) and key$1 and encrypt$4 and interna adj2 node$1 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2004/06/25 10:41
S7	10	compromis$5 and ("leaf node" or leaf adj2 node$1) and key$1 and encrypt$4 and internal adj2 node$1 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 11:34

S8	10	compromis$5 same (leaf or node$) and ("leaf node" or leaf adj2 node$1) and key$1 and internal adj2 node$1 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2004/06/25 11:48
S9	12	compromis$5 same (leaf or node$) and ("leaf node" or leaf adj2 node$1) and internal adj2 node$1 and @ad<"20011214"	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/09/28 11:35

Terms used **define root node** and **hierarchy** Found **51,590** of **161,645**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 Extending object-oriented systems with roles

Georg Gottlob, Michael Schrefl, Brigitte Röck

July 1996 **ACM Transactions on Information Systems (TOIS)**, Volume 14 Issue 3

Full text available: pdf(610.03 KB) Additional Information: full citation, abstract, references, citings, index terms, review

In many class-based object-oriented systems the association between as instance and a class is exclusive and permanent. Therefore these systems have serious difficulties in representing objects taking on different roles over time. Such objects must be reclassified any time they evolve (e.g., if a person becomes a student and later an employee). Class hierarchies must be planned carefully and may grow exponentially if entities may take on serveral independent roles. The problem is even more ...

2 Index-driven similarity search in metric spaces

Gisli R. Hjaltason, Hanan Samet

December 2003 **ACM Transactions on Database Systems (TODS)**, Volume 28 Issue 4

Full text available: pdf(650.64 KB) Additional Information: full citation, abstract, references, citings, index terms

Similarity search is a very important operation in multimedia databases and other database applications involving complex objects, and involves finding objects in a data set *S* similar to a query object *q*, based on some similarity measure. In this article, we focus on methods for similarity search that make the general assumption that similarity is represented with a distance metric *d*. Existing methods for handling similarity search in this setting typically fall into one of ...

Keywords: Hiearchical metric data structures, distance-based indexing, nearest neighbor queries, range queries, ranking, similarity searching

3 Structured hypertext with domain semantics

Weigang Wang, Roy Rada

October 1998 **ACM Transactions on Information Systems (TOIS)**, Volume 16 Issue 4

Full text available: pdf(593.99 KB) Additional Information: full citation, abstract, references, citings, index terms

One important facet of current hypertext research involves using knowledge-based techniques to develop and maintain document structures. A semantic net is one such

technique. However, most semantic-net-based hypertext systems leave the linking consistency of the net to individual users. Users without guidance may accidentally introduce structural and relational inconsistencies in the semantic nets. The relational inconsistency hinders the creation of domain information models. The structura ...

Keywords: graph theory, hypertext models, hypertext structures

4 A comparison of application-level and router-assisted hierarchical schemes for reliable multicast

Pavlin Radoslavov, Christos Papadopoulos, Ramesh Govindan, Deborah Estrin
June 2004 **IEEE/ACM Transactions on Networking (TON)**, Volume 12 Issue 3

Full text available: pdf(725.81 KB) Additional Information: full citation, abstract, references, index terms

One approach to achieving scalability in reliable multicast is to use a hierarchy. A hierarchy can be established at the application level, or by using router-assist. With router-assist we have more fine-grain control over the placement of error-recovery functionality, therefore, a hierarchy produced by assistance from the routers is expected to have better performance. In this paper, we test this hypothesis by comparing two schemes, one that uses an application-level hierarchy (ALH) and another ...

Keywords: reliable multicast, router-assist for reliable multicast

5 Hierarchical packet fair queueing algorithms

Jon C. R. Bennett, Hui Zhang
October 1997 **IEEE/ACM Transactions on Networking (TON)**, Volume 5 Issue 5

Full text available: pdf(455.41 KB) Additional Information: full citation, references, citings, index terms, review

Keywords: hierarchical packet scheduling, link-sharing, quality of service, real-time, resource management

6 A nested transaction model for multilevel secure database management systems

Elisa Bertino, Barbara Catania, Elena Ferrari
November 2001 **ACM Transactions on Information and System Security (TISSEC)**, Volume 4 Issue 4

Full text available: pdf(560.96 KB) Additional Information: full citation, abstract, references, index terms

This article presents an approach to concurrency control for transactions in a Multilevel Secure Database Management System (MLS/DBMS). The major problem is that concurrency control mechanisms used in traditional DBMSs are not adequate in a MLS/DBMS, since they may be exploited to establish covert channels. The approach presented in this article, which uses single-version data items, is based on the use of nested transactions, application-level recovery, and notification-based locking protocols. ...

Keywords: Nested transactions, concurrency control, covert channels, multilevel secure database management systems

7 Exploiting hierarchical domain structure to compute similarity

Prasanna Ganesan, Hector Garcia-Molina, Jennifer Widom
January 2003 **ACM Transactions on Information Systems (TOIS)**, Volume 21 Issue 1

Additional Information: full citation, abstract, references, citings, index

Full text available: pdf(285.80 KB) terms

The notion of similarity between objects finds use in many contexts, for example, in search engines, collaborative filtering, and clustering. Objects being compared often are modeled as sets, with their similarity traditionally determined based on set intersection. Intersection-based measures do not accurately capture similarity in certain domains, such as when the data is sparse or when there are known relationships between items within sets. We propose new measures that exploit a hierarchical ...

Keywords: Similarity measures, collaborative filtering, data mining, hierarchy

8 Dynamic control of performance monitoring on large scale parallel systems

Jeffrey K. Hollingsworth, Barton P. Miller

August 1993 **Proceedings of the 7th international conference on Supercomputing**

Full text available: pdf(1.24 MB) Additional Information: full citation, abstract, references, citings, index terms

Performance monitoring of large scale parallel computers creates a dilemma: we need to collect detailed information to find performance bottlenecks, yet collecting all this data can introduce serious data collection bottlenecks. At the same time, users are being inundated with volumes of complex graphs and tables that require a performance expert to interpret. We present a new approach called the W3 Search Model, that addresses both these problems by combining dynamic o ...

9 Visualization: Query, analysis, and visualization of hierarchically structured data using Polaris

Chris Stolte, Diane Tang, Pat Hanrahan

July 2002 **Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining**

Full text available: pdf(10.02 MB) Additional Information: full citation, abstract, references, citings, index terms

In the last several years, large OLAP databases have become common in a variety of applications such as corporate data warehouses and scientific computing. To support interactive analysis, many of these databases are augmented with hierarchical structures that provide meaningful levels of abstraction that can be leveraged by both the computer and analyst. This hierarchical structure generates many challenges and opportunities in the design of systems for the query, analysis, and visualization of ...

10 Hierarchical planarity testing algorithms

Thomas Lengauer

July 1989 **Journal of the ACM (JACM)**, Volume 36 Issue 3

Full text available: pdf(2.55 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Using hierarchical definitions, one can describe very large graphs in small space. The blow-up from the length of the hierarchical description to the size of the graph can be as large as exponential. If the efficiency of graph algorithms is measured in terms of the length of the hierarchical description rather than in terms of the graph size, algorithms that do not exploit the hierarchy become hopelessly inefficient. Whether the hierarchy can be exploited to speed up the solution of graph p ...

11 Creating generative models from range images

Ravi Ramamoorthi, James Arvo

July 1999 **Proceedings of the 26th annual conference on Computer graphics and interactive techniques**

Full text available: pdf(8.54 MB) Additional Information: full citation, references, citings, index terms

Keywords: curves and surfaces, generative models, procedural modeling, range images

12 Structural analysis of hypertexts: identifying hierarchies and useful metrics

Rodrigo A. Botafogo, Ehud Rivlin, Ben Shneiderman

April 1992 **ACM Transactions on Information Systems (TOIS)**, Volume 10 Issue 2

Full text available: pdf(2.24 MB) Additional Information: full citation, abstract, references, citings, index terms

Hypertext users often suffer from the "lost in hyperspace" problem: disorientation from too many jumps while traversing a complex network. One solution to this problem is improved authoring to create more comprehensible structures. This paper proposes several authoring tools, based on hypertext structure analysis. In many hypertext systems authors are encouraged to create hierarchical structures, but when writing, the hierarchy is lost because of the inclusion of cros ...

Keywords: graph theory, hierarchies, hypertext, metrics, structural analysis

13 SaveMe: a system for archiving electronic documents using messaging groupware

Stefan Berchtold, Alexandros Biliris, Euthimios Panagos

March 1999 **ACM SIGSOFT Software Engineering Notes , Proceedings of the international joint conference on Work activities coordination and collaboration**, Volume 24 Issue 2

Full text available: pdf(1.47 MB) Additional Information: full citation, abstract, references, index terms

Today, organizations deal with an ever-increasing number of documents that have to be archived because they are either related to their core business (e.g., product designs) or needed to meet corporate or legal retention requirements (e.g., voucher). In this paper, we present the architecture and prototype implementation of SaveMe, a document archival system that is based on network-centric groupware such as Internet standards-based messaging systems. In SaveMe, the actions of archiving, retriev ...

Keywords: Internet, archiving, groupware, messaging

14 What makes a good user interface pattern language?

E. Todd, E. Kemp, C. Phillips

January 2004 **Proceedings of the fifth conference on Australasian user interface - Volume 28 CRPIT '04**

Full text available: pdf(502.58 KB) Additional Information: full citation, abstract, references

A developer of user interfaces (UI) should be able to employ a user interface pattern language to design acceptable user interfaces. But, what makes a good pattern language? Three types of validation were identified as requiring consideration: the validity of the individual patterns, the internal validation of the pattern language and the external validation of the pattern language. This paper investigates internal validity. A set of six tests that a developer can use to test the internal validi ...

Keywords: pattern language validation, pattern languages, user interface design

15 A survey of Web metrics

Devanshu Dhyani, Wee Keong Ng, Sourav S. Bhowmick
December 2002 **ACM Computing Surveys (CSUR)**, Volume 34 Issue 4

Full text available: pdf(289.28 KB) Additional Information: full citation, abstract, references, citings, index terms

The unabated growth and increasing significance of the World Wide Web has resulted in a flurry of research activity to improve its capacity for serving information more effectively. But at the heart of these efforts lie implicit assumptions about "quality" and "usefulness" of Web resources and services. This observation points towards measurements and models that quantify various attributes of web sites. The science of measuring all aspects of information, especially its storage and retrieval or ...

Keywords: Information theoretic, PageRank, Web graph, Web metrics, Web page similarity, quality metrics

16 Default representation in constraint-based frameworks

Alex Lascarides, Ann Copestake
March 1999 **Computational Linguistics**, Volume 25 Issue 1

Full text available: pdf(3.17 MB) Publisher Site Additional Information: full citation, abstract, references, citings

Default unification has been used in several linguistic applications. Most of them have utilized defaults at a metalevel, as part of an extended description language. We propose that allowing default unification to be a fully integrated part of a typed feature structure system requires default unification to be a binary, order independent function, so that it acquires the perspicuity and declarativity familiar from normal unification-based frameworks. Furthermore, in order to respect the behavio ...

17 Implementing the product automation formalism

Frederick J. Portier
December 1987 **Proceedings of the 19th conference on Winter simulation**

Full text available: pdf(842.97 KB) Additional Information: full citation, abstract, references, index terms

A formalism for discrete simulation is a set of conventions for the construction of discrete simulation models. In this paper we define the product automaton formalism. The formalism is defined as a mathematical object and is independent of any programming language. Implementing the formalism in a specific programming language entails associating the various components of the formalism with language constructs. We show how the formalism can be implemented in any procedural language. Further ...

18 Conceptual learning in database design

Yannis E. Ionnidis, Tomas Saulys, Andrew J. Whitsitt
July 1992 **ACM Transactions on Information Systems (TOIS)**, Volume 10 Issue 3

Full text available: pdf(2.00 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper examines the idea of incorporating machine learning algorithms into a database system for monitoring its stream of incoming queries and generating hierarchies with the most important concepts expressed in those queries. The goal is for these hierarchies to provide valuable input to the database administrator for dynamically modifying the physical and external schemas of a database for improved system performance and user productivity. The criteria for choosing the appropriate lea ...

Keywords: /UNIMEM, COBWEB, adaptive database systems, learning from examples

19 Employing hierarchical federation communities in the virtual ship architecture

Anthony Cramp, Michael Oudshoorn

January 2002 **Australian Computer Science Communications , Proceedings of the twenty-fifth Australasian conference on Computer science - Volume 4 CRPITS '02**, Volume 24 Issue 1

Full text available: pdf(838.26 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper discusses work underway to develop a framework for the use of hierarchical federation communities as a tool for distributed simulation. The Virtual Ship Project is the application driving the development of the framework. The specific problem within the Virtual Ship Project is one of having to filter unwanted data. It is expected that a hierarchical federation community structure will implicitly provide the necessary data filtering.There are two main goals in establishing hierarchical ...

Keywords: HLA, federation communities, federations of federations, hierarchical federations, virtual ship

20 Cryptographic tools: ID-based encryption for complex hierarchies with applications to forward security and broadcast encryption

Danfeng Yao, Nelly Fazio, Yevgeniy Dodis, Anna Lysyanskaya

October 2004 **Proceedings of the 11th ACM conference on Computer and communications security**

Full text available: pdf(220.00 KB) Additional Information: full citation, abstract, references, index terms

A forward-secure encryption scheme protects secret keys from exposure by evolving the keys with time. Forward security has several unique requirements in hierarchical identity-based encryption (HIBE) scheme: (1) users join dynamically; (2) encryption is joining-time-oblivious; (3) users evolve secret keys autonomously.

We present a scalable forward-secure HIBE (fs-HIBE) scheme satisfying the above properties. We also show how our fs-HIBE scheme can be used to construct a forward-secure ...

Keywords: ID-Based encryption, broadcast encryption, forward security

Terms used **define root node** and **leaf node** and **position** Found **62,190** of **161,645**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 The complexity of XPath query evaluation and XML typing

Georg Gottlob, Christoph Koch, Reinhard Pichler, Luc Segoufin

March 2005 **Journal of the ACM (JACM)**, Volume 52 Issue 2

Full text available: pdf(447.53 KB) Additional Information: full citation, abstract, references, index terms

We study the complexity of two central XML processing problems. The first is XPath 1.0 query processing, which has been shown to be in PTIME in previous work. We prove that both the data complexity and the query complexity of XPath 1.0 fall into lower (highly parallelizable) complexity classes, while the combined complexity is PTIME-hard. Subsequently, we study the sources of this hardness and identify a large and practically important fragment of XPath 1.0 for which the combined complexity is L ...

Keywords: Complexity, DTD, LOGCFL, XML, XPath

2 A needed narrowing strategy

Sergio Antoy, Rachid Echahed, Michael Hanus

July 2000 **Journal of the ACM (JACM)**, Volume 47 Issue 4

Full text available: pdf(336.29 KB) Additional Information: full citation, abstract, references, citings, index terms

The narrowing relation over terms constitutes the basis of the most important operational semantics of languages that integrate functional and logic programming paradigms. It also plays an important role in the definition of some algorithms of unification modulo equational theories that are defined by confluent term rewriting systems. Due to the inefficiency of simple narrowing, many refined narrowing strategies have been proposed in the last decade. This paper presents a new narrowing stra ...

Keywords: call-by-need, functional logic programming languages, narrowing strategies, rewrite systems

3 The Quadtree and Related Hierarchical Data Structures

Hanan Samet

June 1984 **ACM Computing Surveys (CSUR)**, Volume 16 Issue 2

Full text available: pdf(4.87 MB) Additional Information: full citation, references, citings, index terms

4 Optimal evaluation of array expressions on massively parallel machines
Siddhartha Chatterjee, John R. Gilbert, Robert Schreiber, Shang-Hua Teng
January 1995 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 17 Issue 1

Full text available: pdf(2.18 MB) Additional Information: full citation, abstract, references, citings, index terms

We investigate the problem of evaluating Fortran 90-style array expressions on massively parallel distributed-memory machines. On such a machine, an elementwise operation can be performed in constant time for arrays whose corresponding elements are in the same processor. If the arrays are not aligned in this manner, the cost of aligning them is part of the cost of evaluating the expression tree. The choice of where to perform the operation then affects this cost.We describe the c ...

Keywords: Fortran 90, array alignment, compact dynamic programming, data-parallel programming, distributed memory parallel processors, fixed topology Steiner tree

5 Geometric compression through topological surgery
Gabriel Taubin, Jarek Rossignac
April 1998 **ACM Transactions on Graphics (TOG)**, Volume 17 Issue 2

Full text available: pdf(8.98 MB) Additional Information: full citation, abstract, references, citings, index terms

The abundance and importance of complex 3-D data bases in major industry segments, the affordability of interactive 3-D rendering for office and consumer use, and the exploitation of the Internet to distribute and share 3-D data have intensified the need for an effective 3-D geometric compression technique that would significantly reduce the time required to transmit 3-D models over digital communication channels, and the amount of memory or disk space required to store the models. Because ...

Keywords: 3D mesh compression, VRML, geometry compression

6 Session 4B: Algorithms for combining rooted triplets into a galled phylogenetic network
Jesper Jansson, Nguyen Bao Nguyen, Wing-Kin Sung
January 2005 **Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms**

Full text available: pdf(1.06 MB) Additional Information: full citation, abstract, references

This paper considers the problem of determining whether a given set T of rooted triplets can be merged without conflicts into a galled phylogenetic network, and if so, constructing such a network. When the input T is dense, we solve the problem in $O(|T|)$ time, which is optimal since the size of the input is $\Theta(|T|)$. In comparison, the previously fastest algorithm for this problem runs in $O(|T|^2)$ time. Next, we prove that the proble ...

7 Modular stratification and magic sets for Datalog programs with negation
Kenneth A. Ross
November 1994 **Journal of the ACM (JACM)**, Volume 41 Issue 6

Full text available: pdf(3.60 MB) Additional Information: full citation, abstract, references, citings, index terms, review

A class of "modularly stratified" logic programs is defined. Modular stratification generalizes stratification and local stratification, while allowing programs that are not expressible as stratified programs. For modularly stratified programs, the well-founded semantics coincides with the stable model semantics and makes every ground literal true or false.

Modularly stratified programs are weakly stratified, but the converse is false. Unlike some weakly stratified programs, mod ...

Keywords: deductive databases, magic sets, modular stratification, rule rewriting, stratification, well-sounded semantics

8 Typing and querying XML documents: some complexity bounds 

Luc Segoufin

June 2003 **Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems**

Full text available: pdf(246.23 KB) Additional Information: full citation, abstract, references, citings, index terms

We study the complexity bound of validating XML documents, viewed as labeled unranked ordered trees, against various typing systems like DTDs, XML schemas, tree automata ... We also consider query evaluation complexities for various fragments of XPath. For both problems, validation and query evaluation, we consider data and combined complexity bounds.

9 The string B-tree: a new data structure for string search in external memory and its applications 

Paolo Ferragina, Roberto Grossi

March 1999 **Journal of the ACM (JACM)**, Volume 46 Issue 2

Full text available: pdf(363.37 KB) Additional Information: full citation, abstract, references, citings, index terms

We introduce a new text-indexing data structure, the String B-Tree, that can be seen as a link between some traditional external-memory and string-matching data structures. In a short phrase, it is a combination of B-trees and Patricia tries for internal-node indices that is made more effective by adding extra pointers to speed up search and update operations. Consequently, the String B-Tree overcomes the theoretical limitations of inverted files, B-trees, prefix B-trees, s ...

Keywords: B-tree, Patricia trie, external-memory data structure, prefix and range search, string searching and sorting, suffix array, suffix tree, text index

10 Optimal tree layout (Preliminary Version) 

Michael J. Fischer, Michael S. Paterson

April 1980 **Proceedings of the twelfth annual ACM symposium on Theory of computing**

Full text available: pdf(747.81 KB) Additional Information: full citation, abstract, references, citings, index terms

We consider the problem of finding a minimal cost layout of a tree in Euclidian d-space. A tree is an acyclic undirected edge-weighted graph, and a layout is an assignment of a point in d-dimensional Euclidian space to each of the nodes of the tree. The "length" of an edge in the layout is the "distance" between its endpoints as measured by some norm. The cost of an edge is its length times its weig ...

11 A compressed accessibility map for XML 

Ting Yu, Divesh Srivastava, Laks V. S. Lakshmanan, H. V. Jagadish

June 2004 **ACM Transactions on Database Systems (TODS)**, Volume 29 Issue 2

Full text available: pdf(528.00 KB) Additional Information: full citation, abstract, references, index terms

XML is the undisputed standard for data representation and exchange. As companies transact business over the Internet, letting authorized customers directly access, and even

modify, XML data offers many advantages in terms of cost, accuracy, and timeliness. Given the complex business relationships between companies, and the sensitive nature of information, access must be provided selectively, using sophisticated access control specifications. Using the specification directly to determine if a us ...

Keywords: Access control, XML, structural locality

12 Index-driven similarity search in metric spaces

Gisli R. Hjaltason, Hanan Samet

December 2003 **ACM Transactions on Database Systems (TODS)**, Volume 28 Issue 4

Full text available: pdf(650.64 KB) Additional Information: full citation, abstract, references, citings, index terms

Similarity search is a very important operation in multimedia databases and other database applications involving complex objects, and involves finding objects in a data set *S* similar to a query object *q*, based on some similarity measure. In this article, we focus on methods for similarity search that make the general assumption that similarity is represented with a distance metric *d*. Existing methods for handling similarity search in this setting typically fall into one of ...

Keywords: Hiearchical metric data structures, distance-based indexing, nearest neighbor queries, range queries, ranking, similarity searching

13 Necessary and sufficient conditions to linearize doubly recursive programs in logic databases

Weining Zhang, Clement T. Yu, Daniel Troy

September 1990 **ACM Transactions on Database Systems (TODS)**, Volume 15 Issue 3

Full text available: pdf(1.90 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Linearization of nonlinear recursive programs is an important issue in logic databases for both practical and theoretical reasons. If a nonlinear recursive program can be transformed into an equivalent linear recursive program, then it may be computed more efficiently than when the tranformation is not possible. We provide a set of necessary and sufficient conditions for a simple doubly recursive program to be equivalent to a simple linear recursive program. The necessary and sufficient con ...

Keywords: logic database

14 The hB $^\Pi$-tree: a multi-attribute index supporting concurrency, recovery and node consolidation

Georgios Evangelidis, David Lomet, Betty Salzberg

February 1997 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 6 Issue 1

Full text available: pdf(314.15 KB) Additional Information: full citation, abstract, citings, index terms

We propose a new multi-attribute index. Our approach combines the hB-tree, a multi-attribute index, and the Π-tree, an abstract index which offers efficient concurrency and recovery methods. We call the resulting method the hB $^\Pi$-tree. We describe several versions of the hB $^\Pi$-tree, each using a different node-splitting and index-term-posting algorithm. We also describe a new node deletion algorithm. We have implemented all the versions of the hB $^\Pi$-tree. Our performance results ...

Keywords: Concurrency, Multi-attribute index, Node consolidation, Recovery

15 A model parametric real-time logic
Angelo Morzenti, Dino Mandrioli, Carlo Ghezzi
October 1992 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 14 Issue 4

Full text available: pdf(3.54 MB) Additional Information: full citation, abstract, references, citings, index terms, review

TRIO is a formal notation for the logic-based specification of real-time systems. In this paper the language and its straightforward model-theoretic semantics are briefly summarized. Then the need for assigning a consistent meaning to TRIO specifications is discussed, with reference to a variety of underlying time structures such as infinite-time structures (both dense and discrete) and finite-time structures. The main motivation is the ability to validate formal specifications. A solution ...

Keywords: first-order logic, formal specifications, model-theoretic semantics, real-time systems, requirements validation, temporal logic

16 On the efficiency of pairing heaps and related data structures
Michael L. Fredman
July 1999 **Journal of the ACM (JACM)**, Volume 46 Issue 4

Full text available: pdf(191.73 KB) Additional Information: full citation, abstract, references, citings, index terms, review

The pairing heap is well regarded as an efficient data structure for implementing priority queue operations. It is included in the GNU C++ library. Strikingly simple in design, the pairing heap data structure nonetheless seems difficult to analyze, belonging to the genre of self-adjusting data structures. With its design originating as a self-adjusting analogue of the Fibonacci heap, it has been previously conjectured that the pairing heap provides constrant amortized time decrease-key ope ...

Keywords: Fibonacci heaps, amortized complexity analysis, lower bounds, priority queues, self-adjusting data structures

17 Session 9A: Fully-dynamic two dimensional orthogonal range and line segment intersection reporting in logarithmic time
Christian Worm Mortensen
January 2003 **Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms**

Full text available: pdf(1.12 MB) Additional Information: full citation, abstract, references, index terms

We consider the two dimensional fully-dynamic orthogonal range reporting problem and the two dimensional fully-dynamic orthogonal line segment intersection reporting problem in the comparison model. We show that if n is the number of stored elements, then these problems can be solved in worst case time $\Theta(\log n)$ plus time proportional to the size of the output pr. operation.

18 Object-based and image-based object representations
Hanan Samet
June 2004 **ACM Computing Surveys (CSUR)**, Volume 36 Issue 2

Full text available: pdf(1.05 MB) Additional Information: full citation, abstract, references, index terms

An overview is presented of object-based and image-based representations of objects by their interiors. The representations are distinguished by the manner in which they can be

used to answer two fundamental queries in database applications: (1) Feature query: given an object, determine its constituent cells (i.e., their locations in space). (2) Location query: given a cell (i.e., a location in space), determine the identity of the object (or objects) of which it is a member as well as the re ...

Keywords: Access methods, R-trees, feature query, geographic information systems (GIS), image space, location query, object space, octrees, pyramids, quadtrees, space-filling curves, spatial databases

19 Design of an external schema facility to define and process recursive structures

Eric K. Clemons

June 1981 **ACM Transactions on Database Systems (TODS)**, Volume 6 Issue 2

Full text available: pdf(1.08 MB) Additional Information: full citation, abstract, references, citings, index terms

The role of the external schema is to support user views of data and thus to provide programmers with easier data access. This author believes that an external schema facility is best based on hierarchies, both simple and recursive. After a brief introduction to an external schema facility to support simple hierarchical user views, the requirements for a facility for recursive hierarchies are listed and the necessary extensions to the external schema definition language are offered. < ...

Keywords: ANSI SPARC architectures, external schemata, recursive data structures, user views

20 Phylogenetic Networks: Modeling, Reconstructibility, and Accuracy

Bernard M. E. Moret, Luay Nakhleh, Tandy Warnow, C. Randal Linder, Anna Tholse, Anneke Padolina, Jerry Sun, Ruth Timme

January 2004 **IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)**, Volume 1 Issue 1

Full text available: pdf(1.14 MB) Additional Information: full citation

Keywords: Index Terms- Phylogenetic networks, reticulate evolution, error metric, Robinson-Foulds, bipartitions, tripartitions.

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Terms used **leaf node** and **keys** and **non leaf node** Found **41,750** of **161,645**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

☐ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 Effect of node size on the performance of cache-conscious B±-trees

Richard A. Hankins, Jignesh M. Patel

June 2003 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems**, Volume 31 Issue 1

Full text available: pdf(271.16 KB) Additional Information: full citation, abstract, references, index terms

In main-memory databases, the number of processor cache misses has a critical impact on the performance of the system. Cache-conscious indices are designed to improve performance by reducing the number of processor cache misses that are incurred during a search operation. Conventional wisdom suggests that the index's node size should be equal to the cache line size in order to minimize the number of cache misses and improve performance. As we show in this paper, this design choice ignores additi ...

Keywords: B+-tree, cache-conscious, index

2 Performance of B± tree concurrency control algorithms

V. Srinivasan, Michael J. Carey

October 1993 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 2 Issue 4

Full text available: pdf(2.67 MB) Additional Information: full citation, abstract, references, citings

A number of algorithms have been proposed to access B+-trees concurrently, but they are not well understood. In this article, we study the performance of various B+-tree concurrency control algorithms using a detailed simulation model of B+-tree operations in a centralized DBMS. Our study covers a wide range of data contention situations and resource conditions. In addition, based on the performance of the set of B+-tree concurrency control algorithms, ...

Keywords: B+-tree structures, data contention, lock modes, performance, resource conditions, simulation models, workload parameters

3 On zone-balancing of peer-to-peer networks: analysis of random node join

Xiaoming Wang, Yueping Zhang, Xiafeng Li, Dmitri Loguinov

June 2004 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the joint international conference on Measurement and modeling of computer systems**, Volume 32 Issue 1

Full text available: pdf(282.76 KB) Additional Information: full citation, abstract, references, citings, index terms

Balancing peer-to-peer graphs, including zone-size distributions, has recently become an important topic of peer-to-peer (P2P) research [1], [2], [6], [19], [31], [36]. To bring analytical understanding into the various peer-join mechanisms, we study how zone-balancing decisions made during the initial sampling of the peer space affect the resulting zone sizes and derive several asymptotic results for the maximum and minimum zone sizes that hold with high probability.

Keywords: balls-into-bins, load-balancing, modeling, peer-to-peer

4 A new representation for linear lists

Leo J. Guibas, Edward M. McCreight, Michael F. Plass, Janet R. Roberts

May 1977 **Proceedings of the ninth annual ACM symposium on Theory of computing**

Full text available: pdf(831.46 KB) Additional Information: full citation, abstract, references, citings, index terms

We present a new data structure for maintaining a set of records in a linear list according to their key values. This data structure has the property that we can keep a number of fingers at points of interest in the key space (e.g., the beginning or the end of the list), so that access and modification in the neighborhood of a finger is very efficient. In the Section 2 we discuss the general structure of our B-tree. Since we propose to search the tree from a leaf ...

5 Research sessions: implementation techniques: Fractal prefetching B+-Trees: optimizing both cache and disk performance

Shimin Chen, Phillip B. Gibbons, Todd C. Mowry, Gary Valentin

June 2002 **Proceedings of the 2002 ACM SIGMOD international conference on Management of data**

Full text available: pdf(1.49 MB) Additional Information: full citation, abstract, references, citings, index terms

B+-Trees have been traditionally optimized for I/O performance with disk pages as tree nodes. Recently, researchers have proposed new types of B+-Trees optimized for CPU cache performance in main memory environments, where the tree node sizes are one or a few cache lines. Unfortunately, due primarily to this large discrepancy in optimal node sizes, existing disk-optimized B+-Trees suffer from poor cache performance while cache-optimized B+-Trees exhibi ...

6 Industrial sessions: commercial implementation techniques: A compact B-tree

Peter Bumbulis, Ivan T. Bowman

June 2002 **Proceedings of the 2002 ACM SIGMOD international conference on Management of data**

Full text available: pdf(825.30 KB) Additional Information: full citation, abstract, references, index terms

In this paper we describe a Patricia tree-based B-tree variant suitable for OLTP. In this variant, each page of the B-tree contains a local Patricia tree instead of the usual sorted array of keys. It has been implemented in iAnywhere ASA Version 8.0. Preliminary experience has shown that these indexes can provide significant space and performance benefits over existing ASA indexes.

7 Improving index performance through prefetching

Shimin Chen, Phillip B. Gibbons, Todd C. Mowry

May 2001 **ACM SIGMOD Record , Proceedings of the 2001 ACM SIGMOD international conference on Management of data**, Volume 30 Issue 2

Full text available: pdf(322.21 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper proposes and evaluate *Prefetching B+-Trees* (pB+-Trees), which use prefetching to accelerate two important operations on B+-Tree indices: searches and range scans. To accelerate searches, pB+-Trees use prefetching to effectively create wider nodes than the natural data transfer size: e.g., eight vs. one cache lines or disk pages. These wider nodes reduce the height of the B+-Tree, thereby decreasing the number of e ...

8 Physical storage structures: On B-trees: routing schemes and concurrency
Y. S. Kwong, D. Wood
May 1980 **Proceedings of the 1980 ACM SIGMOD international conference on Management of data**
Full text available: pdf(532.63 KB) Additional Information: full citation, abstract, references

In recent years B-trees have become a common data structure for representing large data dictionaries. In this paper we investigate the often ignored relationship between routing schemes and the permissible operations in leaf-search B-trees. We show that among the many commonly used routing schemes in the literature, some are cleaner than others, making them more attractive for leaf-search B-trees supporting concurrency.

9 Fully persistent B+-trees
Sitaram Lanka, Eric Mays
April 1991 **ACM SIGMOD Record , Proceedings of the 1991 ACM SIGMOD international conference on Management of data**, Volume 20 Issue 2
Full text available: pdf(1.07 MB) Additional Information: full citation, references, citings, index terms

10 Research sessions: compression: Dwarf: shrinking the PetaCube
Yannis Sismanis, Antonios Deligiannakis, Nick Roussopoulos, Yannis Kotidis
June 2002 **Proceedings of the 2002 ACM SIGMOD international conference on Management of data**
Full text available: pdf(1.38 MB) Additional Information: full citation, abstract, references, citings, index terms

Dwarf is a highly compressed structure for computing, storing, and querying data cubes. Dwarf identifies prefix and suffix structural redundancies and factors them out by coalescing their store. Prefix redundancy is high on dense areas of cubes but suffix redundancy is significantly higher for sparse areas. Putting the two together fuses the exponential sizes of high dimensional full cubes into a dramatically condensed data structure. The elimination of suffix redundancy has an equally dramatic ...

11 An empirical comparison of B-trees, compact B-trees and multiway trees
David M. Arnow, Aaron M. Tenenbaum
June 1984 **ACM SIGMOD Record , Proceedings of the 1984 ACM SIGMOD international conference on Management of data**, Volume 14 Issue 2
Full text available: pdf(968.89 KB) Additional Information: full citation, abstract, references

It is well-known that the B-tree data structure yields excellent worst-case search costs and for that reason is widely employed in the organization of external files and in the implementation of data bases. In this paper, we examine general B-trees empirically and compare them with a less restrictive structure, the general multiway tree, and a more restrictive structure, the compact B-tree. We compare search costs, insertion costs, and space costs of these three structures for both small and lar ...

12 Secure key agreement for group communications

Wen-Her Yang, Shiuh-Pyng Shieh
November 2001 **International Journal of Network Management**, Volume 11 Issue 6

Full text available: pdf(174.68 KB) Additional Information: full citation, abstract, references, index terms

A secure key agreement protocol for group communications is proposed in this paper, which ensures the authenticity of group members and the privacy of group messages, and provides the properties of perfect forward and backward privacy. In a group session, the common key is collaboratively established by all participants, hence the overhead of key agreement is balanced among group members.

13 The LHAM log-structured history data access method 
Peter Muth, Patrick O'Neil, Achim Pick, Gerhard Weikum
February 2000 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 8 Issue 3-4

Full text available: pdf(494.76 KB) Additional Information: full citation, abstract, index terms

Numerous applications such as stock market or medical information systems require that both historical and current data be logically integrated into a temporal database. The underlying access method must support different forms of "time-travel" queries, the migration of old record versions onto inexpensive archive media, and high insertion and update rates. This paper presents an access method for transaction-time temporal data, called the log-structured history data access method (L ...

Keywords: Data warehouses, Index structures, Performance, Storage systems, Temporal databases

14 Efficient concurrency control in multidimensional access methods 
Kaushik Chakrabarti, Sharad Mehrotra
June 1999 **ACM SIGMOD Record , Proceedings of the 1999 ACM SIGMOD international conference on Management of data**, Volume 28 Issue 2

Full text available: pdf(1.79 MB) Additional Information: full citation, abstract, references, citings, index terms

The importance of multidimensional index structures to numerous emerging database applications is well established. However, before these index structures can be supported as access methods (AMs) in a "commercial-strength" database management system (DBMS), efficient techniques to provide transactional access to data via the index structure must be developed. Concurrent accesses to data via index structures introduce the problem of protecting ranges specified in the retrieval fr ...

15 The design of ¬ 1NF relational databases into nested normal form 
Mark A. Roth, Henry F. Korth
December 1987 **ACM SIGMOD Record , Proceedings of the 1987 ACM SIGMOD international conference on Management of data**, Volume 16 Issue 3

Full text available: pdf(1.51 MB) Additional Information: full citation, abstract, references, citings, index terms

We develop new algorithms for the design of non first normal form relational databases that are in nested normal form. Previously, a set of given multivalued dependencies and those multivalued dependencies implied by given functional dependencies were used to obtain a nested normal form decomposition of a scheme. This method ignored the semantic distinction between functional and multivalued dependencies and utilized only full multivalued dependencies in the design process. We propose new a ...

16 All searches are divided into three parts: string searches using ternary trees 
David E. Siegel

July 1998 **ACM SIGAPL APL Quote Quad , Proceedings of the APL98 conference on Array processing language**, Volume 29 Issue 3

Full text available: pdf(702.49 KB) Additional Information: full citation, abstract, references, index terms

This paper considers the problem of searching for strings in a dictionary or symbol table. It presents a data structure which can be used for this purpose---the Ternary Tree. It considers the theoretical properties of this structure, compared with other possible structures for the same purpose. It presents an implementation of this structure in APL, including code to do a variety of operations on it.

Keywords: data structure, dictionary search, radix search, search algorithms, search tree, symbol table, ternary tree

17 Concurrency and recovery for index trees

David Lomet, Betty Salzberg

August 1997 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 6 Issue 3

Full text available: pdf(168.36 KB) Additional Information: full citation, abstract, citings, index terms

Although many suggestions have been made for concurrency in B $^+$-trees, few of these have considered recovery as well. We describe an approach which provides high concurrency while preserving well-formed trees across system crashes. Our approach works for a class of index trees that is a generalization of the B $^{\rm link}$-tree. This class includes some multi-attribute indexes and temporal indexes. Structural changes in an index tree are decomposed into a sequence of atomic actions, each one ...

Keywords: Access methods, B-trees, Concurrency, Indexing, Recovery

18 On effective multi-dimensional indexing for strings

H. V. Jagadish, Nick Koudas, Divesh Srivastava

May 2000 **ACM SIGMOD Record , Proceedings of the 2000 ACM SIGMOD international conference on Management of data**, Volume 29 Issue 2

Full text available: pdf(1.15 MB) Additional Information: full citation, abstract, references, citings, index terms

As databases have expanded in scope from storing purely business data to include XML documents, product catalogs, e-mail messages, and directory data, it has become increasingly important to search databases based on wild-card string matching: prefix matching, for example, is more common (and useful) than exact matching, for such data. In many cases, matches need to be on multiple attributes/dimensions, with correlations between the dimensions. Traditional multi-dimensional index structures, ...

19 Speeding up construction of PMR quadtree-based spatial indexes

Gisli R. Hjaltason, Hanan Samet

October 2002 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 11 Issue 2

Full text available: pdf(355.72 KB) Additional Information: full citation, abstract, citings, index terms

Spatial indexes, such as those based on the quadtree, are important in spatial databases for efficient execution of queries involving spatial constraints, especially when the queries involve spatial joins. In this paper we present a number of techniques for speeding up the construction of quadtree-based spatial indexes, specifically the PMR quadtree, which can index arbitrary spatial data. We assume a quadtree implementation using the "linear quadtree", a disk-resident representation ...

Keywords: Bulk-loading, I/O, Spatial indexing

20 Spatial indexing of high-dimensional data based on relative approximation

Yasushi Sakurai, Masatoshi Yoshikawa, Shunsuke Uemura, Haruhiko Kojima

October 2002 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 11 Issue 2

Full text available: pdf(216.44 KB) Additional Information: full citation, abstract, index terms

We propose a novel index structure, the A-tree (approximation tree), for similarity searches in high-dimensional data. The basic idea of the A-tree is the introduction of virtual bounding rectangles (VBRs) which contain and approximate MBRs or data objects. VBRs can be represented quite compactly and thus affect the tree configuration both quantitatively and qualitatively. First, since tree nodes can contain a large number of VBR entries, fanout becomes large, which increases search speed. More ...

Keywords: High-dimensional data, Relative approximation, Similarity search

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Terms used **leaf node** and **keys** and **child node** Found **42,201** of **161,645**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 Effect of node size on the performance of cache-conscious B+-trees

Richard A. Hankins, Jignesh M. Patel

June 2003 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems**, Volume 31 Issue 1

Full text available: pdf(271.16 KB) Additional Information: full citation, abstract, references, index terms

In main-memory databases, the number of processor cache misses has a critical impact on the performance of the system. Cache-conscious indices are designed to improve performance by reducing the number of processor cache misses that are incurred during a search operation. Conventional wisdom suggests that the index's node size should be equal to the cache line size in order to minimize the number of cache misses and improve performance. As we show in this paper, this design choice ignores additi ...

Keywords: B+-tree, cache-conscious, index

2 Making B+- trees cache conscious in main memory

Jun Rao, Kenneth A. Ross

May 2000 **ACM SIGMOD Record , Proceedings of the 2000 ACM SIGMOD international conference on Management of data**, Volume 29 Issue 2

Full text available: pdf(406.75 KB) Additional Information: full citation, abstract, references, citings, index terms

Previous research has shown that cache behavior is important for main memory index structures. Cache conscious index structures such as Cache Sensitive Search Trees (CSS-Trees) perform lookups much faster than binary search and T-Trees. However, CSS-Trees are designed for decision support workloads with relatively static data. Although B+-Trees are more cache conscious than binary search and T-Trees, their utilization of a cache line is low since half of the space is used to store ...

3 Concurrency and recovery for index trees

David Lomet, Betty Salzberg

August 1997 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 6 Issue 3

Full text available: pdf(168.36 KB) Additional Information: full citation, abstract, citings, index terms

Although many suggestions have been made for concurrency in B $^+$-trees, few of these have considered recovery as well. We describe an approach which provides high

concurrency while preserving well-formed trees across system crashes. Our approach works for a class of index trees that is a generalization of the B $^{\rm link}$-tree. This class includes some multi-attribute indexes and temporal indexes. Structural changes in an index tree are decomposed into a sequence of atomic actions, each one ...

Keywords: Access methods, B-trees, Concurrency, Indexing, Recovery

4 Group updates for relaxed height-balanced trees
Lauri Malmi, Eljas Soisalon-Soininen
May 1999 **Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems**
Full text available: pdf(1.19 MB) Additional Information: full citation, references, citings, index terms

5 Improving index performance through prefetching
Shimin Chen, Phillip B. Gibbons, Todd C. Mowry
May 2001 **ACM SIGMOD Record , Proceedings of the 2001 ACM SIGMOD international conference on Management of data**, Volume 30 Issue 2
Full text available: pdf(322.21 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper proposes and evaluate *Prefetching B+-Trees* (pB+-Trees), which use prefetching to accelerate two important operations on B+-Tree indices: searches and range scans. To accelerate searches, pB+-Trees use prefetching to effectively create wider nodes than the natural data transfer size: e.g., eight vs. one cache lines or disk pages. These wider nodes reduce the height of the B+-Tree, thereby decreasing the number of e ...

6 A symmetric concurrent B-tree algorithm
Vladimir Lanin, Dennis Shasha
November 1986 **Proceedings of 1986 ACM Fall joint computer conference**
Full text available: pdf(1.10 MB) Additional Information: full citation, references, citings, index terms

7 The string B-tree: a new data structure for string search in external memory and its applications
Paolo Ferragina, Roberto Grossi
March 1999 **Journal of the ACM (JACM)**, Volume 46 Issue 2
Full text available: pdf(363.37 KB) Additional Information: full citation, abstract, references, citings, index terms

We introduce a new text-indexing data structure, the String B-Tree, that can be seen as a link between some traditional external-memory and string-matching data structures. In a short phrase, it is a combination of B-trees and Patricia tries for internal-node indices that is made more effective by adding extra pointers to speed up search and update operations. Consequently, the String B-Tree overcomes the theoretical limitations of inverted files, B-trees, prefix B-trees, s ...

Keywords: B-tree, Patricia trie, external-memory data structure, prefix and range search, string searching and sorting, suffix array, suffix tree, text index

8 Industrial sessions: commercial implementation techniques: A compact B-tree
Peter Bumbulis, Ivan T. Bowman

June 2002 **Proceedings of the 2002 ACM SIGMOD international conference on Management of data**

Full text available: pdf(825.30 KB) Additional Information: full citation, abstract, references, index terms

In this paper we describe a Patricia tree-based B-tree variant suitable for OLTP. In this variant, each page of the B-tree contains a local Patricia tree instead of the usual sorted array of keys. It has been implemented in iAnywhere ASA Version 8.0. Preliminary experience has shown that these indexes can provide significant space and performance benefits over existing ASA indexes.

9 Comparison of access methods for time-evolving data

Betty Salzberg, Vassilis J. Tsotras

June 1999 **ACM Computing Surveys (CSUR)**, Volume 31 Issue 2

Full text available: pdf(529.53 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper compares different indexing techniques proposed for supporting efficient access to temporal data. The comparison is based on a collection of important performance criteria, including the space consumed, update processing, and query time for representative queries. The comparison is based on worst-case analysis, hence no assumptions on data distribution or query frequencies are made. When a number of methods have the same asymptotic worst-case behavior, features in the methods tha ...

Keywords: I/O performance, access methods, structures, temporal databases

10 Research sessions: implementation techniques: Fractal prefetching B$^+$-Trees: optimizing both cache and disk performance

Shimin Chen, Phillip B. Gibbons, Todd C. Mowry, Gary Valentin

June 2002 **Proceedings of the 2002 ACM SIGMOD international conference on Management of data**

Full text available: pdf(1.49 MB) Additional Information: full citation, abstract, references, citings, index terms

B$^+$-Trees have been traditionally optimized for I/O performance with disk pages as tree nodes. Recently, researchers have proposed new types of B$^+$-Trees optimized for CPU cache performance in main memory environments, where the tree node sizes are one or a few cache lines. Unfortunately, due primarily to this large discrepancy in optimal node sizes, existing disk-optimized B$^+$-Trees suffer from poor cache performance while cache-optimized B$^+$-Trees exhibi ...

11 Burst tries: a fast, efficient data structure for string keys

April 2002 **ACM Transactions on Information Systems (TOIS)**, Volume 20 Issue 2

Full text available: pdf(324.84 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Many applications depend on efficient management of large sets of distinct strings in memory. For example, during index construction for text databases a record is held for each distinct word in the text, containing the word itself and information such as counters. We propose a new data structure, the burst trie, that has significant advantages over existing options for such applications: it uses about the same memory as a binary search tree; it is as fast as a trie; and, while not as fast as a ...

Keywords: Binary trees, splay trees, string data structures, text databases, tries, vocabulary accumulation

12 Pipelining with futures
Guy E. Blelloch, Margaret Reid-Miller
June 1997 **Proceedings of the ninth annual ACM symposium on Parallel algorithms and architectures**
Full text available: pdf(1.73 MB) Additional Information: full citation, references, citings, index terms

13 Concurrency and recovery in generalized search trees
Marcel Kornacker, C. Mohan, Joseph M. Hellerstein
June 1997 **ACM SIGMOD Record , Proceedings of the 1997 ACM SIGMOD international conference on Management of data**, Volume 26 Issue 2
Full text available: pdf(1.59 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents general algorithms for concurrency control in tree-based access methods as well as a recovery protocol and a mechanism for ensuring repeatable read. The algorithms are developed in the context of the Generalized Search Tree (GiST) data structure, an index structure supporting an extensible set of queries and data types. Although developed in a GiST context, the algorithms are generally applicable to many tree-based access methods. The concurrency control protocol is base ...

14 The performance of current B-tree algorithms
Theodore Johnson, Dennis Sasha
March 1993 **ACM Transactions on Database Systems (TODS)**, Volume 18 Issue 1
Full text available: pdf(2.87 MB) Additional Information: full citation, references, citings, index terms, review

Keywords: B-trees, concurrent B-trees, concurrent data structures, performance of concurrent algorithms

15 P-trees: storage efficient multiway trees
David M. Arnow, Aaron M. Tenenbaum, Connie Wu
June 1985 **Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval**
Full text available: pdf(471.02 KB) Additional Information: full citation, abstract, references

A new variation of high order multiway tree structures, the P-tree is presented. P-trees have average access costs that are significantly better than those of B-trees and are no worse (and often better) in storage utilization. Unlike compact B-trees, they can be maintained dynamically, and unlike dense multiway trees and B-trees, their associated insertion algorithm, which is also presented, is cheap and involves (at most) a very localized rearrangement of keys.

16 Main-memory index structures with fixed-size partial keys
Philip Bohannon, Peter McIlroy, Rajeev Rastogi
May 2001 **ACM SIGMOD Record , Proceedings of the 2001 ACM SIGMOD international conference on Management of data**, Volume 30 Issue 2
Full text available: pdf(185.51 KB) Additional Information: full citation, abstract, references, citings, index terms

The performance of main-memory index structures is increasingly determined by the number of CPU cache misses incurred when traversing the index. When keys are stored indirectly, as is standard in main-memory databases, the cost of key retrieval in terms of cache misses can dominate the cost of an index traversal. Yet it is inefficient in both time

and space to store even moderate sized keys directly in index nodes. In this paper, we investigate the performance of tree structures suitable for ...

Keywords: B-trees, T-tree, cache coherence, key compression, main-memory indices

17 Optimizing multidimensional index trees for main memory access
Kihong Kim, Sang K. Cha, Keunjoo Kwon
May 2001 **ACM SIGMOD Record , Proceedings of the 2001 ACM SIGMOD international conference on Management of data**, Volume 30 Issue 2
Full text available: pdf(243.75 KB) Additional Information: full citation, abstract, references, citings, index terms

Recent studies have shown that cache-conscious indexes such as the CSB+-tree outperform conventional main memory indexes such as the T-tree. The key idea of these cache-conscious indexes is to eliminate most of child pointers from a node to increase the fanout of the tree. When the node size is chosen in the order of the cache block size, this pointer elimination effectively reduces the tree height, and thus improves the cache behavior of the index. However, the pointer elimination cannot be ...

18 Algorithms and data structures: Concurrent cache-oblivious b-trees
Michael A. Bender, Jeremy T. Fineman, Seth Gilbert, Bradley C. Kuszmaul
July 2005 **Proceedings of the 17th annual ACM symposium on Parallelism in algorithms and architectures SPAA'05**
Full text available: pdf(180.51 KB) Additional Information: full citation, abstract, references, index terms

This paper presents concurrent cache-oblivious (CO) B-trees. We extend the cache-oblivious model to a parallel or distributed setting and present three concurrent CO B-trees. Our first data structure is a concurrent lock-based exponential CO B-tree. This data structure supports insertions and non-blocking searches/successor queries. The second and third data structures are lock-based and lock-free variations, respectively, on the packed-memory CO B-tree. These data structures support range queri ...

Keywords: cache-oblivious b-tree, concurrent b-tree, exponential tree, lock free, non-blocking, packed-memory array

19 Uncoupling updating and rebalancing in chromatic binary search trees
Otto Nurmi, Eljas Soisalon-Soininen
April 1991 **Proceedings of the tenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems**
Full text available: pdf(724.54 KB) Additional Information: full citation, references, citings, index terms

20 The hB-tree: a multiattribute indexing method with good guaranteed performance
David B. Lomet, Betty Salzberg
December 1990 **ACM Transactions on Database Systems (TODS)**, Volume 15 Issue 4
Full text available: pdf(2.58 MB) Additional Information: full citation, abstract, references, citings, index terms, review

A new multiattribute index structure called the hB-tree is introduced. It is derived from the K-D-B-tree of Robinson [15] but has additional desirable properties. The hB-tree internode search and growth processes are precisely analogous to the corresponding processes in B-trees [1]. The intranode processes are unique. A k-d tree is used as the structure within nodes for very efficient searching. Node splitting requires that this k-d tree be split. This produces nodes which no longer represe ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Results for **"(define root node and hierarchy<ln>metadata)"** e-mail
Your search matched **0** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

» Search Options

View Session History

New Search

Modify Search

(define root node and hierarchy<in>metadata)

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer to the Help pages if you need assistan search.

Results for "(root node and hierarchy<in>metadata)" e-mail
Your search matched **129** of **1239820** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

» Search Options

View Session History

New Search

Modify Search

(root node and hierarchy<in>metadata)

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

Select **Article Information** **View: 1-25** | 26-5

☐ **1. A study of the scalability and performance of multi-level hierarchy for sc management wireless IP networks**
Onwuka, E.N.; Niu, Z.;
Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th Proceedings on
Volume 1, 7-10 Sept. 2003 Page(s):465 - 469 Vol.1
Digital Object Identifier 10.1109/PIMRC.2003.1264315
AbstractPlus | Full Text: PDF(526 KB) **IEEE CNF**

☐ **2. Efficient collision detection using bounding volume hierarchies of k-DOP**
Klosowski, J.T.; Held, M.; Mitchell, J.S.B.; Sowizral, H.; Zikan, K.;
Visualization and Computer Graphics, IEEE Transactions on
Volume 4, Issue 1, Jan.-March 1998 Page(s):21 - 36
Digital Object Identifier 10.1109/2945.675649
AbstractPlus | References | Full Text: PDF(1848 KB) **IEEE JNL**

☐ **3. Catalog integration for electronic commerce through category-hierarchy technique**
Dongkyu Kim; Jaebum Kim; Sang-goo Lee;
Research Issues in Data Engineering: Engineering E-Commerce/E-Business S RIDE-2EC 2002. Proceedings. Twelfth International Workshop on
24-25 Feb. 2002 Page(s):28 - 33
Digital Object Identifier 10.1109/RIDE.2002.995095
AbstractPlus | Full Text: PDF(414 KB) **IEEE CNF**

☐ **4. A decomposition algorithm for optimal static load balancing in tree hierar configurations**
Jie Li; Kameda, H.;
Parallel and Distributed Systems, IEEE Transactions on
Volume 5, Issue 5, May 1994 Page(s):540 - 548
Digital Object Identifier 10.1109/71.282565
AbstractPlus | Full Text: PDF(740 KB) **IEEE JNL**

☐ **5. Join index hierarchy: an indexing structure for efficient navigation in obj databases**
Jiawen Han; Zhaohui Xie; Yongjian Fu;
Knowledge and Data Engineering, IEEE Transactions on
Volume 11, Issue 2, March-April 1999 Page(s):321 - 337

Digital Object Identifier 10.1109/69.761666

AbstractPlus | References | Full Text: PDF(564 KB) **IEEE JNL**

☐ **6. Integrating HTML tables using semantic hierarchies and meta-data sets**
Seung-Jin Lim; Yiu-Kai Ng; Xiaochun Yang;
Database Engineering and Applications Symposium, 2002. Proceedings. Inter
17-19 July 2002 Page(s):160 - 169
Digital Object Identifier 10.1109/IDEAS.2002.1029668

AbstractPlus | Full Text: PDF(901 KB) **IEEE CNF**

☐ **7. Clustering a DAG for CAD databases**
Banerjee, J.; Kim, W.; Kim, S.-J.; Garza, J.F.;
Software Engineering, IEEE Transactions on
Volume 14, Issue 11, Nov. 1988 Page(s):1684 - 1699
Digital Object Identifier 10.1109/32.9055

AbstractPlus | Full Text: PDF(1404 KB) **IEEE JNL**

☐ **8. Hierarchy embedded differential image for progressive transmission usin compression**
Whoi-Yul Kim; Balsara, P.T.; Harper, D.T., III; Jon Wong Park;
Circuits and Systems for Video Technology, IEEE Transactions on
Volume 5, Issue 1, Feb. 1995 Page(s):1 - 13
Digital Object Identifier 10.1109/76.350773

AbstractPlus | Full Text: PDF(988 KB) **IEEE JNL**

☐ **9. A text mining approach on automatic generation of Web directories and h**
Hsin-Chang Yang; Chung-Hong Lee;
Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Confer
13-17 Oct. 2003 Page(s):625 - 628

AbstractPlus | Full Text: PDF(243 KB) **IEEE CNF**

☐ **10. Hierarchical schemes for curve representation**
Gunther, O.; Dominguez, S.;
Computer Graphics and Applications, IEEE
Volume 13, Issue 3, May 1993 Page(s):55 - 63
Digital Object Identifier 10.1109/38.210492

AbstractPlus | Full Text: PDF(780 KB) **IEEE JNL**

☐ **11. ACID/HNN: clustering hierarchies of neural networks for context-dependi connectionist acoustic modeling**
Fritsch, J.; Finke, M.;
Acoustics, Speech, and Signal Processing, 1998. ICASSP '98. Proceedings of
International Conference on
Volume 1, 12-15 May 1998 Page(s):505 - 508 vol.1
Digital Object Identifier 10.1109/ICASSP.1998.674478

AbstractPlus | Full Text: PDF(412 KB) **IEEE CNF**

☐ **12. Merging strategic and tactical planning in dynamic and uncertain enviror**
Bonissone, P.P.; Dutta, S.; Wood, N.C.;
Systems, Man and Cybernetics, IEEE Transactions on
Volume 24, Issue 6, June 1994 Page(s):841 - 862
Digital Object Identifier 10.1109/21.293504

AbstractPlus | Full Text: PDF(2192 KB) **IEEE JNL**

☐ **13. FUZZ: a fuzzy-based concept formation system that integrates human ca numerical clustering**
Chen, C.L.P.; Lu, Y.;
Systems, Man and Cybernetics, Part B, IEEE Transactions on

Volume 27, Issue 1, Feb. 1997 Page(s):79 - 94
Digital Object Identifier 10.1109/3477.552187

AbstractPlus | References | Full Text: PDF(340 KB) **IEEE JNL**

☐ **14. Rate-distortion optimization of hierarchical displacement fields**
Hao Bi; Wai-Yip Chan;
Circuits and Systems for Video Technology, IEEE Transactions on
Volume 8, Issue 1, Feb. 1998 Page(s):18 - 24
Digital Object Identifier 10.1109/76.660825

AbstractPlus | References | Full Text: PDF(140 KB) **IEEE JNL**

☐ **15. A flexible approach for visual data mining**
Kreuseler, M.; Schumann, H.;
Visualization and Computer Graphics, IEEE Transactions on
Volume 8, Issue 1, Jan.-March 2002 Page(s):39 - 51
Digital Object Identifier 10.1109/2945.981850

AbstractPlus | References | Full Text: PDF(1580 KB) **IEEE JNL**

☐ **16. Key establishment in large dynamic groups using one-way function trees**
Sherman, A.T.; McGrew, D.A.;
Software Engineering, IEEE Transactions on
Volume 29, Issue 5, May 2003 Page(s):444 - 458
Digital Object Identifier 10.1109/TSE.2003.1199073

AbstractPlus | References | Full Text: PDF(1847 KB) **IEEE JNL**

☐ **17. Data reuse analysis technique for software-controlled memory hierarchie**
Issenin, I.; Brockmeyer, E.; Miranda, M.; Dutt, N.;
Design, Automation and Test in Europe Conference and Exhibition, 2004. Proc
Volume 1, 16-20 Feb. 2004 Page(s):202 - 207 Vol.1
Digital Object Identifier 10.1109/DATE.2004.1268849

AbstractPlus | Full Text: PDF(276 KB) **IEEE CNF**

☐ **18. Beamtrees: compact visualization of large hierarchies**
van Ham, F.; van Wijk, J.J.;
Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on
28-29 Oct. 2002 Page(s):93 - 100
Digital Object Identifier 10.1109/INFVIS.2002.1173153

AbstractPlus | Full Text: PDF(632 KB) **IEEE CNF**

☐ **19. Towards a hierarchical, semantic peer-to-peer topology**
Kurmanowytsch, R.; Jazayeri, M.; Kirda, E.;
Peer-to-Peer Computing, 2002. (P2P 2002). Proceedings. Second Internationa
5-7 Sept. 2002 Page(s):167 - 168
Digital Object Identifier 10.1109/PTP.2002.1046328

AbstractPlus | Full Text: PDF(263 KB) **IEEE CNF**

☐ **20. A hierarchy-based multicast protocol for wireless mobile ad-hoc network**
Tzung-Shi Chen; Yuh-Shyan Chen; Hua-Wen Tsai;
Networks, 2001. Proceedings. Ninth IEEE International Conference on
10-12 Oct. 2001 Page(s):248 - 253
Digital Object Identifier 10.1109/ICON.2001.962348

AbstractPlus | Full Text: PDF(616 KB) **IEEE CNF**

☐ **21. An approach of multi-hierarchy text classification**
Shaohui Liu; Mingkai Dong; Haijun Zhang; Rong Li; Zhongzhi Shi;
Info-tech and Info-net, 2001. Proceedings. ICII 2001 - Beijing. 2001 Internatior
on
Volume 3, 29 Oct.-1 Nov. 2001 Page(s):95 - 100 vol.3

Digital Object Identifier 10.1109/ICII.2001.983042

AbstractPlus | Full Text: PDF(496 KB) **IEEE CNF**

☐ **22. VideoMAP: a generic framework for video management and application p**
Lau, R.W.H.; Qing Li; Si, A.;
System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International C
Jan 4-7 2000 Page(s):10 pp.

AbstractPlus | Full Text: PDF(124 KB) **IEEE CNF**

☐ **23. Automatic generation of sphere hierarchies from CAD data**
Pitt-Francis, J.; Feathertone, R.;
Robotics and Automation, 1998. Proceedings. 1998 IEEE International Confer
Volume 1, 16-20 May 1998 Page(s):324 - 329 vol.1
Digital Object Identifier 10.1109/ROBOT.1998.676414

AbstractPlus | Full Text: PDF(584 KB) **IEEE CNF**

☐ **24. Build-level components**
de Jonge, M.;
Software Engineering, IEEE Transactions on
Volume 31, Issue 7, July 2005 Page(s):588 - 600
Digital Object Identifier 10.1109/TSE.2005.77

AbstractPlus | Full Text: PDF(976 KB) **IEEE JNL**

☐ **25. Hierarchical taxonomy preparation for text categorization using consiste spectral graph copartitioning**
Bin Gao; Tie-Yan Liu; Guang Feng; Tao Qin; Qian-Sheng Cheng; Wei-Ying Ma
Knowledge and Data Engineering, IEEE Transactions on
Volume 17, Issue 9, Sept. 2005 Page(s):1263 - 1273
Digital Object Identifier 10.1109/TKDE.2005.147

AbstractPlus | Full Text: PDF(1024 KB) **IEEE JNL**

View Selected Items

View: 1-25 | 26-5

Results for "(((root node and hierarchy<in>metadata))<and>(root node and leaf node and position<in..." e-mail
Your search matched **3** of **129** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

» Search Options

View Session History

New Search

Modify Search

(((root node and hierarchy<in>metadata))<and>(root node and leaf node and position

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

Select Article Information

☐ **1. Representing hierarchies using multiple synthetic voices**
Shajahan, P.; Irani, P.;
Information Visualisation, 2004. IV 2004. Proceedings. Eighth International Co
14-16 July 2004 Page(s):885 - 890
Digital Object Identifier 10.1109/IV.2004.1320245
AbstractPlus | Full Text: PDF(310 KB) **IEEE CNF**

☐ **2. Interactive and continuous collision detection for avatars in virtual envirc**
Redon, S.; Kim, Y.J.; Lin, M.C.; Manocha, D.; Templeman, J.;
Virtual Reality, 2004. Proceedings. IEEE
27-31 March 2004 Page(s):117 - 283
Digital Object Identifier 10.1109/VR.2004.1310064
AbstractPlus | Full Text: PDF(1068 KB) **IEEE CNF**

☐ **3. A filter coordination for exploring multiclassification sitemaps**
Sifer, M.;
Coordinated and Multiple Views in Exploratory Visualization, 2003. Proceeding
Conference on
15 July 2003 Page(s):112 - 123
AbstractPlus | Full Text: PDF(705 KB) **IEEE CNF**

View Selected Items

Results for **"(leaf node and keys and non-leaf node<in>metadata)"**
Your search matched **0** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

e-mail

» Search Options

View Session History

New Search

Modify Search

(leaf node and keys and non-leaf node<in>metadata)

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer to the Help pages if you need assistan search.

Results for **"((leaf node and keys and child node)<in>metadata)"**
Your search matched **0** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

» **Search Options**

View Session History

New Search

Modify Search

((leaf node and keys and child node)<in>metadata)

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

» **Key**

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer to the Help pages if you need assistan search.

Google

Web Images Groups News Froogle Local more »

define root node and hierarchy Search Advanced Search Preferences

The "AND" operator is unnecessary – we include all search terms by default. [details]

Web Results **1** - **10** of about **987,000** for **define root node and hierarchy**. (**0.33** seconds)

Book results for **define root node and hierarchy**

Programming Dynamic Character Animation - by David Paull - 286 pages
Computer Animation - by Richard Parent - 552 pages

CS 1321X - Lecture 16 - October 9, 2003
The first or **root** element must be a **node** (a number, in this case). ...
tree))))) (**define** (done? tree) (null? tree)) (**define** (found-item? item **root**) (equal? ...
www.cc.gatech.edu/computing/classes/ AY2004/cs1321x_fall/lectures/lec16.html - 21k - Cached - Similar pages

power structure: **Definition** and Much More From Answers.com
A precise, mathematical **definition** of **hierarchy** will be given below. ... as a connected directed acyclic graph with a designated initial **node** (the **root**). ...
www.answers.com/topic/**hierarchy**-1 - 51k - Cached - Similar pages

Copy **Hierarchy**
Any **nodes** or branches that are needed from the original **hierarchy** are taken in ... All TreeLists have a single **root node**.), all the **nodes** should have been ...
www.georeferenceonline.com/TreeListEditor/ WebHelp/Using_TreeList_Editor/Tools/Copy_Hierarchy.htm - 10k - Cached - Similar pages

Bigloo libraries: Generic tree interface
Collect all the **node**-parents and their parents up to **root node** or **nodes**. **Node** self is also ... generic procedure: **node**-next-**hierarchy** self::**node** => **node** ...
bigloo-lib.sourceforge.net/bigloo-lib_8.html - 22k - Cached - Similar pages

[PDF] Efficient Tree Layout in a Multilevel Memory **Hierarchy**
File Format: PDF/Adobe Acrobat - View as HTML
problem for a multilevel memory **hierarchy**. The query performance of ... **node** w is stored in the **root** block (but by **definition** is not in the **root**-block ...
theory.csail.mit.edu/~edemaine/ papers/TreeLayout_ESA2002/paper.pdf - Similar pages

2a. Displaying a DOM **Hierarchy**
Next, **define** the AdapterNode wrapper for DOM **nodes**: ... In this code, the getRoot method returns the **root node** of the DOM, wrapped as an AdapterNode object. ...
java.sun.com/xml/jaxp/dist/ 1.1/docs/tutorial/dom/2_display.html - 40k - Cached - Similar pages

Jetspeed-2 Security Components - Jetspeed 2 Security - **Hierarchy ...**
... <map /> <**node** name="roleid1.1.1"> <map /> </**node**> </**node**> </**node**> </**root**>. This structure would **define** the following group and role **hierarchy**: ...
portals.apache.org/jetspeed-2/ multiproject/jetspeed-security/**hierarchy**.html - 12k - Sep 26, 2005 - Cached - Similar pages

Trilogy :: Patent Portfolio
Logical and Constraint Based Browse **Hierarchy** with Propagation Features ... are related to one another in a tree-like structure starting with a **root node**. ...
www.trilogy.com/patentportfolio.html - 47k - Cached - Similar pages

[PS] CS174b Computer Graphics Bounding Hierarchies Bounding Hierarchies ...

File Format: Adobe PostScript - View as Text
... area of **root node**), it can be removed from the cost calculation. ... We also **define** the cost of a leaf **node** in the **hierarchy** to be zero since it has no ...
www.gg.caltech.edu/~cs174ta/Winter/ Homework/Homework2/boundingHierarchy.ps - Similar pages

Administration Guide
Understanding extension **nodes** · Specific implementations of extension **nodes** and extension **node** adapters ... Appendix A. Managing **root** volume groups ...
www.nersc.gov/vendor_docs/ ibm/pssp/admin/am0a0mst02.html - 28k - Sep 26, 2005 - Cached - Similar pages

Gooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

define root node and hierarchy Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - Hurricane Resources - About Google

Search Results Results **1 - 10** of about **287,000** for **define root node and hierarchy** - 0.21 sec. (Abou

1. Using Yads
   Using Yads. Yads is a complex system, **and** it is not easy to use. Unless your application is really trivial, y
   to **define** the multi-**node** document **hierarchy**, **and** to change the default output ... attribute of your **root** e
   www.twostones.org/soft/yads/using.html - 23k - Cached - More from this site - Save - Block

2. http://www.csd.uwo.ca/~lynda/CS027/2005Jan/ch12_trees.ppt (MICROSOFT POWERPOINT
   ... **define** trees as data structures. **define** the terms associated with ... or predecessor: the **node** directly a
   www.csd.uwo.ca/~lynda/CS027/2005Jan/ch12_trees.ppt - 847k - View as html - More from this site - Sav

3. http://www.tcnj.edu/~mmmartin/CSC230/Slides/ppt09.ppt (MICROSOFT POWERPOINT)
   ... **Define** trees as data structures. **Define** the terms associated ... in which elements are organized into a
   www.tcnj.edu/~mmmartin/CSC230/Slides/ppt09.ppt - 959k - View as html - More from this site - Save - Bl

4. http://www.cclabs.missouri.edu/things/instruction/aw/ReleaseNotes/06_anim.notes.html
   Animation Notes. General Animation Notes. Limitation:Animation Turntable does not rotate about the orig
   with **hierarchy** set to BELOW. ... its inertia tensor, which also happens to **define** the object's principal axe
   cclabs.missouri.edu/things/.../aw/ReleaseNotes/06_anim.notes.html - 61k - Cached - More from this site -

5. Goops Manual
   ... **Node**:Class **hierarchy and** inheritance of slots, Next:Instance creation **and** ... <object>, the **root** of all
   gnu.org/software/.../Class-**hierarchy-and**-inheritance-of-slots.html - 2k - Cached - More from this site - Sa

6. VDS_64_BITS
   ... **Define** VDS_64_BITS for 64-bit **node** paths. A vdsNodePath is a bit vector encapsulating the path fron
   vdslib.virginia.edu/vdsdoc/VDS_64_BITS.html - 2k - Cached - More from this site - Save - Block

7. UNIDO
   UNIDO had the need to support "multiple chapter documents". These have some content themselves, this
   chapter document. ... A tree starts with a **root node** (the document which is at **hierarchy** level 1). So it mi
   www.unido.org/doc/doc/user-**hierarchy** - 39k - Cached - More from this site - Save - Block

8. Class **Hierarchy** Viewer
   The Class **Hierarchy** Viewer. The class **hierarchy** viewer is the viewer that you will be in by default when
   browsing starts at the **root** of the class **hierarchy**. This can be ... may assign a **node** style to a new group
   www.ai.sri.com/~gkb/**hierarchy**.html - 30k - Cached - More from this site - Save - Block

9. Network Security and Architecture Laboratory
   bfs.cc File Reference. Include dependency graph for bfs.cc: **Defines**. Functions. Variables. **Define** Docun
   BFS. const NodeVec_t & N,
   www.ece.gatech.edu/research/labs/nsa/gtnets/html/bfs_8cc.shtml - 39k - Cached - More from this site - S

10. Control API for Tree View (treeView)
    Control API for Tree View (treeView) A representation of hierarchical data (for example, directory **and** file
    of treeNodes **define** the **hierarchy** level. ... **Defines** the **root node** of tree. This attribute is used when the
    www.sapdesignguild.org/resources/htmlb_guidance/treeview_dev.html - 27k - Cached - More from this sit

Results Page:

1 2 3 4 5 6 7 8 9 10 Next

Web | Images | Video | Directory | Local | News | Shopping
Your Search: define root node and hierarchy Search

Search Results Results **1 - 10** of about **60,600** for **define root node and leaf nodes and position** - 0.

1. programming_17
   The tree is one of the most important data structures we will study. ... of **nodes** from the **root node** to the **and** ...
   campus.murraystate.edu/academic/faculty/.../245/programming_17.html - 22k - Cached - More from this s

2. Chapter 3. **Nodes and Node** Types
   Chapter 3. **Nodes and Node** Types. Chapter 3. **Nodes and Node** Types. A scene graph holds the data tl coordinate system contained within it. **Nodes** A, B, **and** C are the **leaf** geometry **nodes** ...
   www.cs.rochester.edu/u/wyi/sgi/ch03.html - 86k - Cached - More from this site - Save - Block

3. Scene Graph Basics
   The Java 3D API Specification. C H A P T E R 3. Scene Graph Basics. A scene graph consists of Java 3I attaches them to a virtual universe. ... **node** object, along the path from the **root** to the **leaf**, that changes
   java.sun.com/products/java-media/.../j3dguide/SceneGraphOverview.html - 28k - Cached - More from this

4. SGI TPL (IRIX 6.5: Developer/Perf_PG - Chapter 3. **Nodes and Node** Types)
   ... **and position** it in ... **root node** of the instanced object, **and** D is a dynamic coordinate system containe
   techpubs.sgi.com/library/tpl/cgi-bin/getdoc.cgi?.../Perf_PG/ch03.html - 103k - Cached - More from this site

5. TopXML : XML in a Nutshell - XPath
   ... XPath indicates **nodes** by **position**, relative **position**, type, content, **and** ... **nodes** contain other **node**
   www.topxml.com/xsl/articles/xpath?printerversion=true - 62k - Cached - More from this site - Save - Block

6. http://pdos.csail.mit.edu/6.824/handouts/l19.txt
   cell phone numbers a value is a phone's current location Basic idea Two layers: routing (lookup) **and** data ... **node** ID assignment 2. **Define** per-**node** routing table ... c Every **node** acts as a **root**, so there's no **ro**
   pdos.csail.mit.edu/6.824/**hand**outs/l19.txt - 6k - Cached - More from this site - Save - Block

7. http://pdos.csail.mit.edu/6.824-2001/lecnotes/l20.txt
   key X goes on **node node** with nearest ID to X Now how, given X, do we find that **node**? Arrange **nodes** acts as a **root**, so there's no **root** ... tables to reflect new **node**. Other **nodes' leaf** sets are ...
   pdos.csail.mit.edu/6.824-2001/lecnotes/l20.txt - 9k - Cached - More from this site - Save - Block

8. http://www.apl.jhu.edu/Classes/605202/felikson/lectures/L6/L6.html
   ... **leaf**) or. an internal **node** (the **root** of the tree) **and** ... **Define**: h = height. n = **# nodes**. ni ... been delet
   www.apl.jhu.edu/Classes/605202/felikson/lectures/L6/L6.html - 33k - Cached - More from this site - Save

9. Dukas User's Guide: 2.1 Building a Heirarchical Task Tree
   ... be difficult to **define** the meaning of ... a subtree, **position** the cursor over the **root** of the ... The **root** n
   www.cc.gatech.edu/gvu/ui/Mastermind/Dukas/doc/ug2.1.html - 5k - Cached - More from this site - Save -

10. http://longwood.cs.ucf.edu/courses/cop3502/fall04/trees1.doc (MICROSOFT WORD)
    ... pointers to **define** a non-linear ... at any **position** of the game ... **Nodes** having no children are called l
    longwood.cs.ucf.edu/courses/cop3502/fall04/trees1.doc - 115k - View as html - More from this site - Save

Results Page:
1 2 3 4 5 6 7 8 9 10 **Next**

Search Results Results **1 - 10** of about **104,000** for **leaf nodes and keys and non-leaf nodes** - 0.13 s

1. http://www-db.stanford.edu/~ullman/dbsi/win99/handouts/hw4.html
   What would happen? ... trees, except that **non-leaf nodes** also have record ... are distinct. **Non-leaf nod**
   www-db.stanford.edu/~ullman/dbsi/win99/h**and**outs/hw4.html - 3k - Cached - More from this site - Save -

2. http://www.cs.umd.edu/class/spring2003/cmsc424-0401/ClassNotesApr10.ppt (MICROSC
   ... search **keys** in **non-leaf nodes** don't appear in the **leaf nodes**, so an additional ... **leaf nodes** look like
   www.cs.umd.edu/class/spring2003/cmsc424-0401/ClassNotesApr10.ppt - 77k - View as html - More from

3. • B+-Tree Overview and some definitions (PDF)
   ... **Leaf nodes** in ... **keys** in **non-leaf nodes** don't appear in the **leaf nodes**, so an. additional pointer field
   www.cs.umd.edu/class/spring2003/cmsc424-0401/ClassNotesApr10.pdf - 58k - View as html - More from

4. http://www.mathcs.sjsu.edu/faculty/lee/cs157b/CS157BL9Indexing%20.ppt (MICROSOFT
   ... **leaf** level. 2) **Non-leaf nodes** store **keys** to help ... 4) All **non-leaf nodes** have between M/2 **and** ... ou
   www.mathcs.sjsu.edu/faculty/lee/cs157b/CS157BL9Indexing .ppt - 717k - View as html - More from this s

5. Indexing (B+ Tree) (PDF)
   ... • Secondary index (non-clustering index) ... the tree balanced, **nodes** should not be too ... distribute **ke**
   www.diku.dk/undervisning/2002f/729/slides/class3.pdf - 88k - View as html - More from this site - Save - E

6. http://www.mtsu.edu/~csjudy/3110/NotesBtrees.doc (MICROSOFT WORD)
   Notes on B-trees of order n. Definition: A B-tree of order n is a structure with these properties: Every **nod**
   nodes in the B-tree. ... **leaf nodes** are on the same level. If a **non-leaf node** has m **keys** ...
   www.mtsu.edu/~csjudy/3110/NotesBtrees.doc - 21k - View as html - More from this site - Save - Block

7. http://www.ics.uci.edu/~ics214a/handouts/slides06.ppt (MICROSOFT POWERPOINT)
   ICS 214A: Database Management Systems. Winter 2004. B-Trees. Professor Sharad Mehrotra. ICS214A
   "balanced." ... to **keys** to **keys** to **keys** to **keys** ... So to store N tuples, N/pl* **leaf nodes** needed. ... has re
   www.ics.uci.edu/~ics214a/h**and**outs/slides06.ppt - 199k - View as html - More from this site - Save - Bloc

8. JDBM JavaDoc: Class BTree
   ... because they store multiple **keys** on one tree **node** ... the **leaf nodes** directly contain (inline) the value
   jdbm.sourceforge.net/doc/javadoc/jdbm/btree/BTree.html - 27k - Cached - More from this site - Save - Blc

9. Database System Technology CSC 443S/03 (PDF)
   ... (b) **leaf** overflow. (c) **non-leaf** overflow ... Avoid duplicate **keys**. – Have record pointers in **non-leaf no**
   www.cs.toronto.edu/db/courses/443/w03/Indexes_B+Trees_2.pdf - 241k - View as html - More from this s

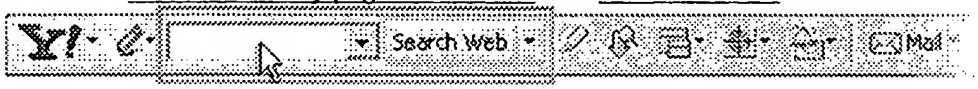
10. http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15451-s99/www/lectures/lect02C
    ... sorted array. * each **non-leaf** has degree = (**# keys**)+1. So, in ... flexibility in sizes **and** degrees of **nod**
    cs.cmu.edu/afs/cs.cmu.edu/academic/class/.../www/lectures/lect0209 - 7k - Cached - More from this site -

Results Page:
1 2 3 4 5 6 7 8 9 10 **Next**

Web | Images | Video | Directory | Local | News | Shopping

Your Search: leaf nodes and keys and non-leaf nodes Search

Search from any page on the Web with Yahoo! Toolbar. It's free



Search Results Results **1 - 10** of about **104,000** for **leaf nodes and keys and non-leaf nodes** - 0.13 s

1. http://www-db.stanford.edu/~ullman/dbsi/win99/handouts/hw4.html
   What would happen? ... trees, except that **non-leaf nodes** also have record ... are distinct. **Non-leaf nod**
   www-db.stanford.edu/~ullman/dbsi/win99/handouts/hw4.html - 3k - Cached - More from this site - Save -

2. http://www.cs.umd.edu/class/spring2003/cmsc424-0401/ClassNotesApr10.ppt (MICROSC
   ... search **keys** in **non-leaf nodes** don't appear in the **leaf nodes**, so an additional ... **leaf nodes** look like
   www.cs.umd.edu/class/spring2003/cmsc424-0401/ClassNotesApr10.ppt - 77k - View as html - More from

3. • B+-Tree Overview and some definitions (PDF)
   ... **Leaf nodes** in ... **keys** in **non-leaf nodes** don't appear in the **leaf nodes**, so an. additional pointer field
   www.cs.umd.edu/class/spring2003/cmsc424-0401/ClassNotesApr10.pdf - 58k - View as html - More from

4. http://www.mathcs.sjsu.edu/faculty/lee/cs157b/CS157BL9Indexing%20.ppt (MICROSOFT
   ... **leaf** level. 2) **Non-leaf nodes** store **keys** to help ... 4) All **non-leaf nodes** have between M/2 **and** ... ou
   www.mathcs.sjsu.edu/faculty/lee/cs157b/CS157BL9Indexing .ppt - 717k - View as html - More from this s

5. Indexing (B+ Tree) (PDF)
   ... • Secondary index (non-clustering index) ... the tree balanced, **nodes** should not be too ... distribute **ke**
   www.diku.dk/undervisning/2002f/729/slides/class3.pdf - 88k - View as html - More from this site - Save - E

6. http://www.mtsu.edu/~csjudy/3110/NotesBtrees.doc (MICROSOFT WORD)
   Notes on B-trees of order n. Definition: A B-tree of order n is a structure with these properties: Every **node**
   nodes in the B-tree. ... **leaf nodes** are on the same level. If a **non-leaf node** has m **keys** ...
   www.mtsu.edu/~csjudy/3110/NotesBtrees.doc - 21k - View as html - More from this site - Save - Block

7. http://www.ics.uci.edu/~ics214a/handouts/slides06.ppt (MICROSOFT POWERPOINT)
   ICS 214A: Database Management Systems. Winter 2004. B-Trees. Professor Sharad Mehrotra. ICS214A
   "balanced." ... to **keys** to **keys** to **keys** to **keys** ... So to store N tuples, N/pl* **leaf nodes** needed. ... has re
   www.ics.uci.edu/~ics214a/handouts/slides06.ppt - 199k - View as html - More from this site - Save - Bloc

8. JDBM JavaDoc: Class BTree
   ... because they store multiple **keys** on one tree **node** ... the **leaf nodes** directly contain (inline) the value
   jdbm.sourceforge.net/doc/javadoc/jdbm/btree/BTree.html - 27k - Cached - More from this site - Save - Blc

9. Database System Technology CSC 443S/03 (PDF)
   ... (b) **leaf** overflow. (c) **non-leaf** overflow ... Avoid duplicate **keys**. – Have record pointers in **non-leaf no**
   www.cs.toronto.edu/db/courses/443/w03/Indexes_B+Trees_2.pdf - 241k - View as html - More from this s

10. http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15451-s99/www/lectures/lect02C
    ... sorted array. * each **non-leaf** has degree = (**# keys**)+1. So, in ... flexibility in sizes **and** degrees of **nod**
    cs.cmu.edu/afs/cs.cmu.edu/academic/class/.../www/lectures/lect0209 - 7k - Cached - More from this site -

Results Page:

1 2 3 4 5 6 7 8 9 10 **Next**

Your Search: leaf nodes and keys and non-leaf nodes Search

Search from any page on the Web with Yahoo! Toolbar. It's free